

Foundation of Cryptography (0368-4162-01), Lecture 2

Pseudorandom Generators

Iftach Haitner, Tel Aviv University

Tel Aviv University.

February 25, 2014

Part I

Statistical Vs. Computational distance

Section 1

Distributions and Statistical Distance

Distributions and Statistical Distance

Let P and Q be two distributions over a finite set \mathcal{U} . Their **statistical distance** (also known as, variation distance) is defined as

$$\text{SD}(P, Q) := \frac{1}{2} \sum_{x \in \mathcal{U}} |P(x) - Q(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (P(\mathcal{S}) - Q(\mathcal{S}))$$

We will only consider **finite** distributions.

Distributions and Statistical Distance

Let P and Q be two distributions over a finite set \mathcal{U} . Their **statistical distance** (also known as, variation distance) is defined as

$$\text{SD}(P, Q) := \frac{1}{2} \sum_{x \in \mathcal{U}} |P(x) - Q(x)| = \max_{\mathcal{S} \subseteq \mathcal{U}} (P(\mathcal{S}) - Q(\mathcal{S}))$$

We will only consider **finite** distributions.

Claim 1

For any pair of (finite) distribution P and Q , it holds that

$$\text{SD}(P, Q) = \max_D \{ \Pr_{x \leftarrow P}[D(x) = 1] - \Pr_{x \leftarrow Q}[D(x) = 1] \},$$

where D is **any** algorithm.

Some useful facts

Let P, Q, R be finite distributions, then

Triangle inequality:

$$\text{SD}(P, R) \leq \text{SD}(P, Q) + \text{SD}(Q, R)$$

Repeated sampling:

$$\text{SD}((P, P), (Q, Q)) \leq 2 \cdot \text{SD}(P, Q)$$

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$ is a **distribution ensemble**, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.

\mathcal{P} is **efficiently samplable** (or just efficient), if \exists PPT *Samp* with $\text{Sam}(1^n) \equiv P_n$.

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$ is a **distribution ensemble**, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.

\mathcal{P} is **efficiently samplable** (or just efficient), if \exists PPT $Samp$ with $Sam(1^n) \equiv P_n$.

Definition 3 (statistical indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **statistically indistinguishable**, if $SD(P_n, Q_n) = \text{neg}(n)$.

Distribution ensembles and statistical indistinguishability

Definition 2 (distribution ensembles)

$\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$ is a **distribution ensemble**, if P_n is a (finite) distribution for any $n \in \mathbb{N}$.

\mathcal{P} is **efficiently samplable** (or just efficient), if \exists PPT $Samp$ with $Sam(1^n) \equiv P_n$.

Definition 3 (statistical indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **statistically indistinguishable**, if $SD(P_n, Q_n) = \text{neg}(n)$.

Alternatively, if $\left| \Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) \right| = \text{neg}(n)$, for *any* algorithm D , where

$$\Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) := \Pr_{x \leftarrow P_n} [D(1^n, x) = 1] - \Pr_{x \leftarrow Q_n} [D(1^n, x) = 1] \quad (1)$$

Section 2

Computational Indistinguishability

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **computationally indistinguishable**, if $\left| \Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) \right| = \text{neg}(n)$, for any **PPT** D .

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **computationally indistinguishable**, if $\left| \Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) \right| = \text{neg}(n)$, for any **PPT** D .

- Can it be different from the statistical case?

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **computationally indistinguishable**, if $\left| \Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) \right| = \text{neg}(n)$, for any **PPT** D .

- Can it be different from the statistical case?
- Non uniform variant

Computational Indistinguishability

Definition 4 (computational indistinguishability)

Two distribution ensembles \mathcal{P} and \mathcal{Q} are **computationally indistinguishable**, if $\left| \Delta_{(\mathcal{P}, \mathcal{Q})}^D(n) \right| = \text{neg}(n)$, for any **PPT** D .

- Can it be different from the statistical case?
- Non uniform variant
- Sometime behaves differently then expected!

Repeated sampling

Question 5

Assume that \mathcal{P} and \mathcal{Q} are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})$ are?

Repeated sampling

Question 5

Assume that \mathcal{P} and \mathcal{Q} are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^2, \mathcal{Q}^2)}^D(n) \right|$

Repeated sampling

Question 5

Assume that \mathcal{P} and \mathcal{Q} are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^2, \mathcal{Q}^2)}^D(n) \right|$

$$\begin{aligned} \delta(n) &= \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \\ &\leq \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] \right| \\ &\quad + \left| \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \end{aligned}$$

Repeated sampling

Question 5

Assume that \mathcal{P} and \mathcal{Q} are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^2, \mathcal{Q}^2)}^D(n) \right|$

$$\begin{aligned} \delta(n) &= \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \\ &\leq \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] \right| \\ &\quad + \left| \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \\ &= \left| \Delta_{(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q}))}^D(n) \right| + \left| \Delta_{((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)}^D(n) \right| \end{aligned}$$

Repeated sampling

Question 5

Assume that \mathcal{P} and \mathcal{Q} are computationally indistinguishable, is it always true that $\mathcal{P}^2 = (\mathcal{P}, \mathcal{P})$ and $\mathcal{Q}^2 = (\mathcal{Q}, \mathcal{Q})$ are?

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^2, \mathcal{Q}^2)}^D(n) \right|$

$$\begin{aligned}\delta(n) &= \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \\ &\leq \left| \Pr_{x \leftarrow \mathcal{P}_n^2} [D(x) = 1] - \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] \right| \\ &\quad + \left| \Pr_{x \leftarrow (\mathcal{P}_n, \mathcal{Q}_n)} [D(x) = 1] - \Pr_{x \leftarrow \mathcal{Q}_n^2} [D(x) = 1] \right| \\ &= \left| \Delta_{(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q}))}^D(n) \right| + \left| \Delta_{((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)}^D(n) \right|\end{aligned}$$

So either $\left| \Delta_{(\mathcal{P}^2, (\mathcal{P}, \mathcal{Q}))}^D(n) \right| \geq \delta(n)/2$, or $\left| \Delta_{((\mathcal{P}, \mathcal{Q}), \mathcal{Q}^2)}^D(n) \right| \geq \delta(n)/2$

- Assume D is a PPT and that $\left| \Delta_{(\mathcal{P}^2, \mathbb{Q}^2)}^D(n) \right| \geq 1/p(n)$ for some $p \in \text{poly}$ and infinitely many n 's, and assume wlg. that $\left| \Delta_{\mathcal{P}^2, (\mathcal{P}, \mathbb{Q})}^D(n) \right| \geq 1/2p(n)$ for infinitely many n 's.

- Assume D is a PPT and that $\left| \Delta_{(\mathcal{P}^2, \mathbb{Q}^2)}^D(n) \right| \geq 1/p(n)$ for some $p \in \text{poly}$ and infinitely many n 's, and assume wlg. that $\left| \Delta_{\mathcal{P}^2, (\mathcal{P}, \mathbb{Q})}^D(n) \right| \geq 1/2p(n)$ for infinitely many n 's.
- Can we use D to contradict the fact that \mathcal{P} and \mathbb{Q} are computationally close?

- Assume D is a PPT and that $\left| \Delta_{(\mathcal{P}^2, \mathbb{Q}^2)}^D(n) \right| \geq 1/p(n)$ for some $p \in \text{poly}$ and infinitely many n 's, and assume wlg. that $\left| \Delta_{\mathcal{P}^2, (\mathcal{P}, \mathbb{Q})}^D(n) \right| \geq 1/2p(n)$ for infinitely many n 's.
- Can we use D to contradict the fact that \mathcal{P} and \mathbb{Q} are computationally close?
- Assuming that \mathcal{P} and \mathbb{Q} are efficiently samplable

- Assume D is a PPT and that $\left| \Delta_{(\mathcal{P}^2, \mathbb{Q}^2)}^D(n) \right| \geq 1/p(n)$ for some $p \in \text{poly}$ and infinitely many n 's, and assume wlg. that $\left| \Delta_{\mathcal{P}^2, (\mathcal{P}, \mathbb{Q})}^D(n) \right| \geq 1/2p(n)$ for infinitely many n 's.
- Can we use D to contradict the fact that \mathcal{P} and \mathbb{Q} are computationally close?
- Assuming that \mathcal{P} and \mathbb{Q} are efficiently samplable
- Non-uniform settings

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n \in \mathbb{N}}$.

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n \in \mathbb{N}}$.

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that \mathcal{P} and \mathcal{Q} are eff. samplable and computationally indistinguishable, does it mean that \mathcal{P}^t and \mathcal{Q}^t are?

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n \in \mathbb{N}}$.

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that \mathcal{P} and \mathcal{Q} are eff. samplable and computationally indistinguishable, does it mean that \mathcal{P}^t and \mathcal{Q}^t are?

Proof:

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n \in \mathbb{N}}$.

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that \mathcal{P} and \mathcal{Q} are eff. samplable and computationally indistinguishable, does it mean that \mathcal{P}^t and \mathcal{Q}^t are?

Proof:

- Induction?

Repeated sampling cont.

Given $t = t(n) \in \mathbb{N}$ and a distribution ensemble $\mathcal{P} = \{P_n\}_{n \in \mathbb{N}}$, let $\mathcal{P}^t = \{P_n^{t(n)}\}_{n \in \mathbb{N}}$.

Question 6

Let $t = t(n) \leq \text{poly}(n)$ be an eff. computable integer function. Assume that \mathcal{P} and \mathcal{Q} are eff. samplable and computationally indistinguishable, does it mean that \mathcal{P}^t and \mathcal{Q}^t are?

Proof:

- Induction?
- Hybrid

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^t, \mathcal{Q}^t)}^D(n) \right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0, \dots, t = t(n)\}$, let $H^i = (p_1, \dots, p_i, q_{i+1}, \dots, q_t)$, where the p 's [resp., q 's] are uniformly (and independently) chosen from P_n [resp., from Q_n].

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^t, \mathcal{Q}^t)}^D(n) \right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0, \dots, t = t(n)\}$, let $H^i = (p_1, \dots, p_i, q_{i+1}, \dots, q_t)$, where the p 's [resp., q 's] are uniformly (and independently) chosen from P_n [resp., from Q_n].
- Since $\delta(n) = \left| \Delta_{H^t, H^0}^D(t) \right| = \left| \sum_{i \in [t]} \Delta_{H^i, H^{i-1}}^D(t) \right|$, there exists $i \in [t]$ with $\left| \Delta_{H^i, H^{i-1}}^D(t) \right| \geq \delta(n)/t(n)$.

Hybrid argument

Let D be an algorithm and let $\delta(n) = \left| \Delta_{(\mathcal{P}^t, \mathcal{Q}^t)}^D(n) \right|$.

- Fix $n \in \mathbb{N}$, and for $i \in \{0, \dots, t = t(n)\}$, let $H^i = (p_1, \dots, p_i, q_{i+1}, \dots, q_t)$, where the p 's [resp., q 's] are uniformly (and independently) chosen from P_n [resp., from Q_n].
- Since $\delta(n) = \left| \Delta_{H^t, H^0}^D(t) \right| = \left| \sum_{i \in [t]} \Delta_{H^i, H^{i-1}}^D(t) \right|$, there exists $i \in [t]$ with $\left| \Delta_{H^i, H^{i-1}}^D(t) \right| \geq \delta(n)/t(n)$.
- How do we use it?

Using hybrid argument via **estimation**

Algorithm 7 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Find $i \in [t]$ with $\left| \Delta_{H^i, H^{i-1}}^D(t) \right| \geq \delta(n)/2t(n)$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t), \dots$

Using hybrid argument via **estimation**

Algorithm 7 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- ➊ Find $i \in [t]$ with $\left| \Delta_{H^i, H^{i-1}}^D(t) \right| \geq \delta(n)/2t(n)$
- ➋ Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- ➌ Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t), \dots$

- ➊ how do we find i ? why $\delta(n)/2t(n)$

Using hybrid argument via estimation

Algorithm 7 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Find $i \in [t]$ with $\left| \Delta_{H^i, H^{i-1}}^D(t) \right| \geq \delta(n)/2t(n)$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t), \dots$

- 1 how do we find i ? why $\delta(n)/2t(n)$
- 2 Easy in the non-uniform case

Using hybrid argument via **sampling**

Algorithm 8 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Sample $i \leftarrow [t = t(n)]$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t)$.

Using hybrid argument via **sampling**

Algorithm 8 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Sample $i \leftarrow [t = t(n)]$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t)$.

$$\left| \Delta_{(\mathcal{P}, \mathbb{Q})}^{D'}(n) \right| = \left| \Pr_{p \leftarrow P_n} [D'(p) = 1] - \Pr_{q \leftarrow Q_n} [D'(q) = 1] \right|$$

Using hybrid argument via **sampling**

Algorithm 8 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Sample $i \leftarrow [t = t(n)]$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t)$.

$$\begin{aligned} \left| \Delta_{(\mathcal{P}, \mathbb{Q})}^{D'}(n) \right| &= \left| \Pr_{p \leftarrow P_n} [D'(p) = 1] - \Pr_{q \leftarrow Q_n} [D'(q) = 1] \right| \\ &= \left| \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_i} [D(x) = 1] - \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_{i-1}} [D(x) = 1] \right| \end{aligned}$$

Using hybrid argument via **sampling**

Algorithm 8 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Sample $i \leftarrow [t = t(n)]$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t)$.

$$\begin{aligned} \left| \Delta_{(\mathcal{P}, \mathbb{Q})}^{D'}(n) \right| &= \left| \Pr_{p \leftarrow P_n} [D'(p) = 1] - \Pr_{q \leftarrow Q_n} [D'(q) = 1] \right| \\ &= \left| \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_i} [D(x) = 1] - \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_{i-1}} [D(x) = 1] \right| \\ &= \left| \frac{1}{t} \left(\Pr_{x \leftarrow H_t} [D(x) = 1] - \Pr_{x \leftarrow H_0} [D(x) = 1] \right) \right| \end{aligned}$$

Using hybrid argument via **sampling**

Algorithm 8 (D')

Input: 1^n and $x \in \{0, 1\}^*$

- 1 Sample $i \leftarrow [t = t(n)]$
- 2 Let $(p_1, \dots, p_i, q_{i+1}, \dots, q_t) \leftarrow H^i$
- 3 Return $D(1^t, p_1, \dots, p_{i-1}, x, q_{i+1}, \dots, q_t)$.

$$\begin{aligned} \left| \Delta_{(\mathcal{P}, \mathbb{Q})}^{D'}(n) \right| &= \left| \Pr_{p \leftarrow P_n} [D'(p) = 1] - \Pr_{q \leftarrow Q_n} [D'(q) = 1] \right| \\ &= \left| \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_i} [D(x) = 1] - \frac{1}{t} \sum_{i \in [t]} \Pr_{x \leftarrow H_{i-1}} [D(x) = 1] \right| \\ &= \left| \frac{1}{t} \left(\Pr_{x \leftarrow H_t} [D(x) = 1] - \Pr_{x \leftarrow H_0} [D(x) = 1] \right) \right| \\ &= \delta(n)/t(n) \end{aligned}$$

Part II

Pseudorandom Generators

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

- Do such distributions exist?

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

- Do such distributions exist?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g : \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a **pseudorandom generator**, if

- ▶ g is length extending (i.e., $\ell(n) > n$ for any n)
- ▶ $g(U_n)$ is pseudorandom

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

- Do such distributions exist?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g : \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a **pseudorandom generator**, if

- ▶ g is length extending (i.e., $\ell(n) > n$ for any n)
- ▶ $g(U_n)$ is pseudorandom

- Do such generators exist?

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

- Do such distributions exist?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g : \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a **pseudorandom generator**, if

- ▶ g is length extending (i.e., $\ell(n) > n$ for any n)
- ▶ $g(U_n)$ is pseudorandom

- Do such generators exist?
- Imply one-way functions (homework)

Pseudorandom generator

Definition 9 (pseudorandom distributions)

A distribution ensemble \mathcal{P} over $\{\{0, 1\}^{\ell(n)}\}_{n \in \mathbb{N}}$ is **pseudorandom**, if it is computationally indistinguishable from $\{U_{\ell(n)}\}_{n \in \mathbb{N}}$.

- Do such distributions exist?

Definition 10 (pseudorandom generators (PRGs))

An efficiently computable function $g : \{0, 1\}^n \mapsto \{0, 1\}^{\ell(n)}$ is a **pseudorandom generator**, if

- ▶ g is length extending (i.e., $\ell(n) > n$ for any n)
- ▶ $g(U_n)$ is pseudorandom

- Do such generators exist?
- Imply one-way functions (homework)
- Do they have any use?

Section 3

Hardcore Predicates

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0, 1\}^n \mapsto \{0, 1\}$ is a **hardcore predicate** of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, if

$$\Pr_{x \leftarrow \{0,1\}^n} [P(f(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P .

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0, 1\}^n \mapsto \{0, 1\}$ is a **hardcore predicate** of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, if

$$\Pr_{x \leftarrow \{0,1\}^n} [P(f(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P .

- Does the existence of a hardcore predicate for f , implies that f is one way?

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0, 1\}^n \mapsto \{0, 1\}$ is a **hardcore predicate** of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, if

$$\Pr_{x \leftarrow \{0,1\}^n} [P(f(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P .

- Does the existence of a hardcore predicate for f , implies that f is one way? If f is injective?

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0, 1\}^n \mapsto \{0, 1\}$ is a **hardcore predicate** of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, if

$$\Pr_{x \leftarrow \{0,1\}^n} [P(f(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P .

- Does the existence of a hardcore predicate for f , implies that f is one way? If f is injective?
- Fact: any PRG has HCP (homework).

Hardcore predicates

- Building blocks in constructions of PRGS from OWF

Definition 11 (hardcore predicates)

An efficiently computable function $b : \{0, 1\}^n \mapsto \{0, 1\}$ is a **hardcore predicate** of $f : \{0, 1\}^n \mapsto \{0, 1\}^n$, if

$$\Pr_{x \leftarrow \{0,1\}^n} [P(f(x)) = b(x)] \leq \frac{1}{2} + \text{neg}(n),$$

for any PPT P .

- Does the existence of a hardcore predicate for f , implies that f is one way? If f is injective?
- Fact: any PRG has HCP (homework).
- Fact: any OWF has a hardcore predicate (next class)

Section 4

PRGs from OWPs

Claim 12

Let $f : \{0, 1\}^n \mapsto \{0, 1\}^n$ be a permutation and let $b : \{0, 1\}^n \mapsto \{0, 1\}$ be a hardcore predicate for f , then $g(x) = (f(x), b(x))$ is a PRG.

Claim 12

Let $f : \{0, 1\}^n \mapsto \{0, 1\}^n$ be a permutation and let $b : \{0, 1\}^n \mapsto \{0, 1\}$ be a hardcore predicate for f , then $g(x) = (f(x), b(x))$ is a PRG.

Proof: Assume \exists a PPT D , and infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in \text{poly}$ with

$$\left| \Delta_{g(U_n), U_{n+1}}^D \right| > \varepsilon(n) = 1/p(n)$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of b .

Claim 12

Let $f : \{0, 1\}^n \mapsto \{0, 1\}^n$ be a permutation and let $b : \{0, 1\}^n \mapsto \{0, 1\}$ be a hardcore predicate for f , then $g(x) = (f(x), b(x))$ is a PRG.

Proof: Assume \exists a PPT D , and infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in \text{poly}$ with

$$\left| \Delta_{g(U_n), U_{n+1}}^D \right| > \varepsilon(n) = 1/p(n)$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of b .

- We assume wlg. that $\Pr[D(g(U_n)) = 1] - \Pr[D(U_{n+1}) = 1] \geq \varepsilon(n)$ for any $n \in \mathcal{I}$ (?), and fix $n \in \mathcal{I}$.

OWP to PRG cont.

- Let $\delta(n) = \Pr[D(U_{n+1}) = 1]$ (note that $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon$).

OWP to PRG cont.

- Let $\delta(n) = \Pr[D(U_{n+1}) = 1]$ (note that $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG cont.

- Let $\delta(n) = \Pr[D(U_{n+1}) = 1]$ (note that $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\ &= \frac{1}{2}(\delta + \varepsilon) + \frac{1}{2} \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].\end{aligned}$$

OWP to PRG cont.

- Let $\delta(n) = \Pr[D(U_{n+1}) = 1]$ (note that $\Pr[D(g(U_n)) = 1] = \delta + \varepsilon$).
- Compute

$$\begin{aligned}\delta &= \Pr[D(f(U_n), U_1) = 1] \\ &= \Pr[U_1 = b(U_n)] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = b(U_n)] \\ &\quad + \Pr[U_1 = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}] \\ &= \frac{1}{2}(\delta + \varepsilon) + \frac{1}{2} \cdot \Pr[D(f(U_n), U_1) = 1 \mid U_1 = \overline{b(U_n)}].\end{aligned}$$

Hence,

$$\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon \quad (2)$$

OWP to PRG cont.

- $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$

OWP to PRG cont.

- $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b :

Algorithm 13 (P)

Input: $y \in \{0, 1\}^n$

- 1 Flip a random coin $c \leftarrow \{0, 1\}$.
- 2 If $D(y, c) = 1$ output c , otherwise, output \bar{c} .

OWP to PRG cont.

- $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b :

Algorithm 13 (P)

Input: $y \in \{0, 1\}^n$

- 1 Flip a random coin $c \leftarrow \{0, 1\}$.
- 2 If $D(y, c) = 1$ output c , otherwise, output \overline{c} .

- It follows that

$$\begin{aligned}\Pr[P(f(U_n)) = b(U_n)] \\ &= \Pr[c = b(U_n)] \cdot \Pr[D(f(U_n), c) = 1 \mid c = b(U_n)] \\ &\quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), c) = 0 \mid c = \overline{b(U_n)}]\end{aligned}$$

OWP to PRG cont.

- $\Pr[D(f(U_n), b(U_n)) = 1] = \delta + \varepsilon$
- $\Pr[D(f(U_n), \overline{b(U_n)}) = 1] = \delta - \varepsilon$
- Consider the following algorithm for predicting b :

Algorithm 13 (P)

Input: $y \in \{0, 1\}^n$

- 1 Flip a random coin $c \leftarrow \{0, 1\}$.
- 2 If $D(y, c) = 1$ output c , otherwise, output \bar{c} .

- It follows that

$$\begin{aligned} & \Pr[P(f(U_n)) = b(U_n)] \\ &= \Pr[c = b(U_n)] \cdot \Pr[D(f(U_n), c) = 1 \mid c = b(U_n)] \\ & \quad + \Pr[c = \overline{b(U_n)}] \cdot \Pr[D(f(U_n), c) = 0 \mid c = \overline{b(U_n)}] \\ &= \frac{1}{2} \cdot (\delta + \varepsilon) + \frac{1}{2}(1 - \delta + \varepsilon) = \frac{1}{2} + \varepsilon. \end{aligned}$$

Remark 14

- Prediction to distinguishing (homework)

Remark 14

- Prediction to distinguishing (homework)
- PRG from any OWF: (1) Regular OWFs, first use pairwise hashing to convert into “almost” permutation. (2) Any OWF, harder

Section 5

PRG Length Extension

PRG Length Extension

Construction 15 (iterated function)

Given $g: \{0, 1\}^n \mapsto \{0, 1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i: \{0, 1\}^n \mapsto \{0, 1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\dots,n+1}),$$

where $g^0(x) = x$.

PRG Length Extension

Construction 15 (iterated function)

Given $g: \{0, 1\}^n \mapsto \{0, 1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i: \{0, 1\}^n \mapsto \{0, 1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\dots,n+1}),$$

where $g^0(x) = x$.

Claim 16

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^{n+1}$ be a PRG, then $g^{t(n)}: \{0, 1\}^n \mapsto \{0, 1\}^{n+t(n)}$ is a PRG, for any $t \in \text{poly}$.

PRG Length Extension

Construction 15 (iterated function)

Given $g: \{0, 1\}^n \mapsto \{0, 1\}^{n+1}$ and $i \in \mathbb{N}$, define $g^i: \{0, 1\}^n \mapsto \{0, 1\}^{n+i}$ as

$$g^i(x) = g(x)_1, g^{i-1}(g(x)_{2,\dots,n+1}),$$

where $g^0(x) = x$.

Claim 16

Let $g: \{0, 1\}^n \mapsto \{0, 1\}^{n+1}$ be a PRG, then $g^{t(n)}: \{0, 1\}^n \mapsto \{0, 1\}^{n+t(n)}$ is a PRG, for any $t \in \text{poly}$.

Proof: Assume \exists a PPT D , an infinite set $\mathcal{I} \subseteq \mathbb{N}$ and $p \in \text{poly}$ with

$$\left| \Delta_{g^t(U_n), U_{n+t(n)}}^D \right| > \varepsilon(n) = 1/p(n),$$

for any $n \in \mathcal{I}$. We use D for breaking the hardness of g .

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \dots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \dots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \dots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D')

Input: 1^n and $y \in \{0, 1\}^{n+1}$

- 1 Sample $i \leftarrow [t]$
- 2 Return $D(1^n, U_{t-i}, y_1, g^{i-1}(y_{2,\dots,n+1}))$.

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \dots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D')

Input: 1^n and $y \in \{0, 1\}^{n+1}$

- 1 Sample $i \leftarrow [t]$
- 2 Return $D(1^n, U_{t-i}, y_1, g^{i-1}(y_2, \dots, y_{n+1}))$.

Claim 18

It holds that $\left| \Delta_{g(U_n), U_{n+1}}^{D'} \right| > \varepsilon(n)/t(n)$

PRG Length Extension cont.

- Fix $n \in \mathbb{N}$, for $i \in \{0, \dots, t = t(n)\}$, let $H^i = U_{t-i}, g^i(U_n)$ (i.e., the distribution of H^i is $(x, g^i(x'))_{x \leftarrow \{0,1\}^{t-i}, x' \leftarrow \{0,1\}^n}$)
- Note that $H^0 \equiv U_{n+t}$ and $H^t \equiv g^t(U_n)$.

Algorithm 17 (D')

Input: 1^n and $y \in \{0, 1\}^{n+1}$

- 1 Sample $i \leftarrow [t]$
- 2 Return $D(1^n, U_{t-i}, y_1, g^{i-1}(y_2, \dots, y_{n+1}))$.

Claim 18

It holds that $\left| \Delta_{g(U_n), U_{n+1}}^{D'} \right| > \varepsilon(n)/t(n)$

Proof: ...